A $(2 - c\frac{\log N}{N})$–Approximation Algorithm for the Stable Marriage Problem

Kazuo Iwama[1⋆], Shuichi Miyazaki[2⋆⋆], and Kazuya Okamoto[1]

[1]Graduate School of Informatics, Kyoto University
[2]Academic Center for Computing and Media Studies, Kyoto University
{iwama, shuichi, okia}@kuis.kyoto-u.ac.jp

Abstract. We propose an approximation algorithm for the problem of finding a maximum stable matching when both ties and unacceptable partners are allowed in preference lists. Our algorithm achieves the approximation ratio $2 - c\frac{\log N}{N}$ for an arbitrarily positive constant c, where N denotes the number of men in an input. This improves the trivial approximation ratio of two.

1 Introduction

The stable marriage problem is a matching problem first introduced by Gale and Shapley [4]. An instance of this problem consists of N men, N women and each person's preference list. A preference list is a totally ordered list including all members of the opposite sex depending on his/her preference. For a matching M between men and women, a pair of a man m and a woman w is called a *blocking pair* if (i) m prefers w to his current partner and (ii) w prefers m to her current partner. A matching with no blocking pair is called *stable*. The stable marriage problem is to find a stable matching for a given instance. Gale and Shapley showed that every instance admits at least one stable matching, and they also proposed so-called the Gale-Shapley algorithm to find one, which runs in $O(N^2)$ time [4].

However, considering an application to a large-scale assignment system, it is unreasonable to force agents to write *all* members of the other party in a *strict* order. Hence two natural relaxations are considered: One is to allow for indifference [6, 11], in which each person is allowed to include *ties* in his/her preference. When ties are allowed, the definition of stability needs to be extended. A man and a woman form a blocking pair if each *strictly* prefers the other to his/her current partner. A matching without such a blocking pair is called *weakly stable* (or simply "stable") and the Gale-Shapley algorithm can be modified to always find a weakly stable matching [6]. The other one is to allow participants to declare one or more *unacceptable partners*. Thus each person's preference list may be incomplete. Again, the definition of a blocking pair is extended, so that each member of the pair prefers the other over the current partner *or* is

⋆ Supported in part by Scientific Research Grant, Ministry of Japan, 13480081 and 15630001
⋆⋆ Supported in part by Scientific Research Grant, Ministry of Japan, 15700010

currently single and acceptable. In this case, a stable matching may not be a perfect matching, but all stable matchings for a fixed instance are of the same size [5]. Hence, finding a maximum cardinality stable matching is trivial.

However, if both ties and incomplete lists are allowed, one instance can admit stable matchings of different sizes, and it is known that the problem of finding a maximum stable matching, which we call *MAX SMTI* (MAXimum Stable Marriage with Ties and Incomplete lists), is NP-hard [14, 17]. For approximability, it is easy to see that two stable matchings for the same instance differ in size by at most a factor of two (see Theorem 5 of [17], for example). Since a stable matching can be found in polynomial time by a modified Gale-Shapley algorithm, existence of an approximation algorithm with a factor of two is trivial. Very recently, [9] shows several approximability upper bounds which are significantly better than two for restricted inputs, such as a factor of $\frac{2}{(1+1/L^2)}$ for instances where length of ties is at most L and ties appear in only one sex.

Our Contribution. In this paper, we give the first nontrivial approximability result for *general* MAX SMTI. Namely, our new algorithm, based on local search, achieves an approximation factor of $2 - c\frac{\log N}{N}$, where c is an arbitrarily positive constant. From an initial stable matching, our algorithm successively improves the size of the solution. While the size of the current solution is at most $\frac{OPT}{2} + c\log N$ where OPT is the size of an optimal solution, we can increase the size by at least one. Hence, we finally obtain a stable matching of size greater than $\frac{OPT}{2} + c\log N$.

Related Results. There are several examples of using the stable marriage problem in assignment systems. Among others, one of the most famous applications is to assign medical students to hospitals based on the preference lists of both sides. For example, more than 30,000 applicants are enrolled in the hospitals/residents matching system in the U.S., which is known as NRMP [6, 16]. In Japan, this kind of matching system came into use since 2003, where more than 95 % of 8,000 applicants obtained their positions in its first year. Other examples are CaRMS in Canada and SPA in Scotland [12, 13]. Another famous application is to assign students to schools in Norway [3] and Singapore [20].

Up to now, there have been a lot of efforts to obtain approximability and inapproximability results for MAX SMTI. For inapproximability, MAX SMTI was shown to be APX-hard [7], and subsequently, a lower bound 21/19 on the approximation ratio (under the assumption that P$\neq$NP) was presented [9]. This lower bound holds for restricted instances where ties appear in only one sex, the length of ties is two, and each person writes at most one tie. For approximability, there are some approximation algorithms with factor better than two for restricted inputs, in which mainly restrictions are done in terms of occurrence of ties and/or lengths of ties [17, 8, 9], as mentioned previously.

There are several optimization problems that resemble MAX SMTI, where designing a 2-approximation algorithm is trivial but obtaining a $(2-\epsilon)$ – approximation algorithm for a positive constant ϵ is extremely hard, such as Minimum Vertex Cover (MIN VC for short) and Minimum Maximal Matching (MIN MM for short). As is the case with MAX SMTI, there are a lot of approximability results for these problems by restricting instances. For example, MIN VC is approximable within 7/6 if the maximum degree of an input graph is bounded by 3 [2], or within $2/(1+\epsilon)$ if every vertex has degree at least $\epsilon|V|$ [15]. For MIN MM, there is a $(2-1/d)$-approximation algorithm for regular graphs with degree d [21], and PTAS for planar graphs [19]. For general inputs, $(2-o(1))$-approximation algorithms are presented for MIN VC, namely, $(2-\frac{\log\log|V|}{2\log|V|})$ and $(2-(1-o(1))\frac{2\ln\ln|V|}{\ln|V|})$ [18, 1, 10].

2 Preliminaries

In this section, we formally define MAX SMTI and approximation ratio of approximation algorithms.

An instance I of MAX SMTI consists of N men, N women and each person's preference list that may be incomplete and may include ties. If a person p writes a person q in his/her list, we say that q is *acceptable* to p. Let m be a man. If m strictly prefers w_i to w_j in I, we write $w_i \succ_m w_j$. If w_i and w_j are tied in m's list, we write $w_i =_m w_j$. The statement $w_i \succeq_m w_j$ is true if and only if $w_i \succ_m w_j$ or $w_i =_m w_j$. We use a similar notation for women's preference lists. Let M be a matching. If a man m is matched with a woman w in M, we write $M(m)=w$ and $M(w)=m$. We say that m and w form a *blocking pair* for M (or simply, (m,w) *blocks* M) if the following three conditions are met: (i) $M(m)\neq w$ but m and w are acceptable to each other. (ii) $w \succ_m M(m)$ or m is single in M. (iii) $m \succ_w M(w)$ or w is single in M. For a matching M, $BP(M)$ denotes the set of all blocking pairs for M. A matching M is called *stable* if and only if $BP(M)=\emptyset$. MAX SMTI is the problem of finding a largest stable matching.

A goodness measure of an approximation algorithm T of a maximization problem is defined as usual: the *approximation ratio* of T is $\max\{opt(x)/T(x)\}$ over all instances x of size N, where $opt(x)$ and $T(x)$ are the size of the optimal and the algorithm's solution, respectively.

3 Overview of Algorithm LocalSearch(I)

Here we give an overview of our algorithm LocalSearch. We need two parameters k and c, which are fixed constants such that $c<\frac{k}{16}$. LocalSearch takes an input I of MAX SMTI and uses two subroutines, Increase and Stabilize.

Increase takes a stable matching M for I and a subset S of M such that $|S| = k\log N$. It outputs a (not necessarily stable) matching M_0 such that $|M_0| > |M|$, and for any blocking pair $(m,w)\in BP(M_0)$, either m or w (or both) is single in M_0. Increase may fail to find such a matching. In such a case, it returns an error.

Stabilize takes a (not necessarily stable) matching M_0 where, for any blocking pair $(m,w)\in BP(M_0)$, either m or w (or both) is single in M_0. It outputs a stable matching of size at least $|M_0|$ (Lemma 10).

The full description of LocalSearch is given in Fig. 1. One can see that application of the while-loop increases the size of stable matching by at least one. This process can continue as long as the condition at line 7 is true. Later, we show that this is the case if (1) an input S for Increase has some "nice" property, and (2) $|M|$, the size of the input stable matching for Increase, is at most $\frac{OPT}{2}+c\log N$ (Lemma 4), where OPT denotes the size of a maximum

Algorithm LocalSearch(I)

1: M: = arbitrary stable matching for I;
  /* This can be done in polynomial time by arbitrary tie-breaking
  and applying the Gale-Shapley algorithm. */
2: **while** (true);
3: {select $(k + 4c) \log N$ edges from M in an arbitrary way,
  and let P be the set of selected edges;
4: let $P_1, P_2, \cdots, P_n$ be all subsets of P of size $k \log N$;
5: **for** $i := 1$ **to** n
6: $M_i :=$ Increase $(M,\ P_i)$;
  /* If Increase returns an error, let M_i be empty. */
7: **if** (there is an M_i such that $|M_i| > |M|$)
8: $M_0 := M_i$;
9: **else**
10: terminate and output M;
11: $M :=$ Stabilize (M_0);
12: }

Fig. 1. Algorithm LocalSearch

stable matching, and c is a constant defined above. Furthermore, we show that, among $P_1, P_2, \cdots, P_n$ obtained at line 4, there is at least one "nice" P_i if $|M| \leq \frac{OPT}{2} + c \log N$ (Lemma 3). Hence, we have the following theorem:

Theorem 1. *Given an SMTI instance I of size N,* LocalSearch *outputs a stable matching of size more than $\frac{OPT}{2} + c \log N$ in time polynomial in N.*

Since constants c and k can be set arbitrarily large, we have the following corollary.

Corollary 1. *For any positive constant c, there is a polynomial-time approximation algorithm for MAX SMTI with approximation ratio at most $2 - c\frac{\log N}{N}$.*

Before showing Increase and Stabilize, we prove an important property of $P_1, P_2, \cdots, P_n$ obtained at line 4 of LocalSearch.

Let us fix an optimal solution M_{opt}, a largest stable matching for I (which we do not know of course). Given a stable matching M for I, let us define the following bipartite graph $G_{M_{opt},M}$: Each vertex of $G_{M_{opt},M}$ corresponds to a person in I. There is an edge between vertices m and w if and only if $M_{opt}(m) = w$ or $M(m) = w$. If both $M_{opt}(m) = w$ and $M(m) = w$ hold, we give two edges between m and w; hence $G_{M_{opt},M}$ is a multigraph. An edge (m, w) associated with $M_{opt}(m) = w$ is called an *OPT-edge*. Similarly, an edge associated with $M(m) = w$ is called an *M-edge*. Observe that the degree of each vertex is at most two, and hence each connected component of $G_{M_{opt},M}$ is a simple path, a cycle or an isolated vertex.

Let us partition M-edges of $G_{M_{opt},M}$ into good edges and bad ones. If an edge is in the path of length three starting from and ending with OPT-edges, then it is called *good*. Otherwise, it is *bad*. We also call an edge in M good (bad, respectively) if that M-edge in $G_{M_{opt},M}$ is good (bad, respectively).

Lemma 1. *Let (m, w) be a good edge of M. Then, $w \succeq_m M_{opt}(m)$ and $m \succeq_w M_{opt}(w)$.*

Proof. If $M_{opt}(m) \succ_m w$, then $(m, M_{opt}(m))$ is a blocking pair for M, which contradicts the stability of M. So, $w \succeq_m M_{opt}(m)$. For the same reason, $m \succeq_w M_{opt}(w)$. □

Lemma 2. *Let t be an arbitrary positive integer. If $|M| \leq \frac{|M_{opt}|}{2} + t$, then the number of bad edges in $G_{M_{opt},M}$ is at most $4t$.*

Proof. First of all, we show that there is no path of length one in $G_{M_{opt},M}$. This can be seen as follows: Suppose that there is a path of length one, say (m, w), and suppose that this is an OPT-edge. Then m and w write each other on the preference list since they are matched in M_{opt}. However, both of them are single in M. This means that (m, w) is a blocking pair for M, which contradicts the stability of M. When (m, w) is an M-edge, we can do a similar argument to have a contradiction.

Consider then each connected component C of $G_{M_{opt},M}$. Let $R(C)$ be the ratio of the number of OPT-edges to the number of M-edges in C. If C is a cycle, then it contains the same number of OPT-edges and M-edges, and hence, $R(C) = 1$. This is same if C is a path of even length. If C is a path of odd length starting from and ending with M-edges, $R(C) < 1$ since the number of M-edges in C is more than that of OPT-edges. If C is a path of length three starting from and ending with OPT-edges, then the M-edge it contains is good and $R(C) = 2$. If C is a path of length more than three starting from and ending with OPT-edges, then $R(C) \leq 3/2$.

Now, suppose that there are ℓ_1 good edges and ℓ_2 bad edges. Then, the number of OPT-edges, namely $|M_{opt}|$ is at most $2\ell_1 + \frac{3}{2}\ell_2$ by the above argument. Since $\ell_1 + \ell_2 = |M|$ and $|M| \leq \frac{|M_{opt}|}{2} + t$, we have that $\ell_2 \leq 4t$. □

Lemma 3. *If $|M| \leq \frac{|M_{opt}|}{2} + c \log N$, then there is at least one i such that P_i contains only good edges.*

Proof. Recall that $|P| = (k + 4c) \log N$. Since there are at most $4c \log N$ bad edges in M as proved in Lemma 2, P contains at least $k \log N$ good edges. Since we output all subsets of size $k \log N$, there must be P_i with only good edges. □

4 Procedure Increase(M, S)

Recall that Increase takes a stable matching M and its subset S of size $k \log N$ as an input, and outputs a matching, say M', where $|M'| > |M|$. M' may not be stable for I but it satisfies the property that for any blocking pair $(m, w) \in BP(M')$, either m or w (or both) is single in M'. Before going to the detail, we roughly explain the execution of Increase.

In the following, we assume that S consists of only good edges. (As proved in Lemma 3, there is one way of receiving such S if $|M| \leq \frac{|M_{opt}|}{2} + c \log N$.) Given S, let S_i be a subset of S whose size is $|S|/4$. Since each edge in S_i is good, for each person p in S_i, his/her partner in M_{opt} is single in M. We divorce all couples of S_i, and then, make them to find a partner who is single in M. They may not find the partner in M_{opt}, but if we try all possible S_i, at least one choice will give us a good result, i.e., every person in S_i finds a partner who is at least as good as the partner in M_{opt} (Lemma 5). Let L be the set of newly added edges. Then, it is not hard to see that $|L| = 2|S_i|$, and hence we can increase the size of M by $|S_i|$. (See Fig. 2 (a).)

In the latter half of the algorithm, we do the following: If there is a blocking pair (m, w) such that both m and w have a partner, say, w' and m', respectively, then, we can prove that exactly one of (m, w') or (m', w) is in L. We then remove one which is *not* in L. (See Fig. 2 (b).) This process may decrease the size of a matching, but we prove that its size decrease is less than $|S_i|$. In total, we can increase the size of matching at least by one. The full description of algorithm Increase is given in Fig. 3.

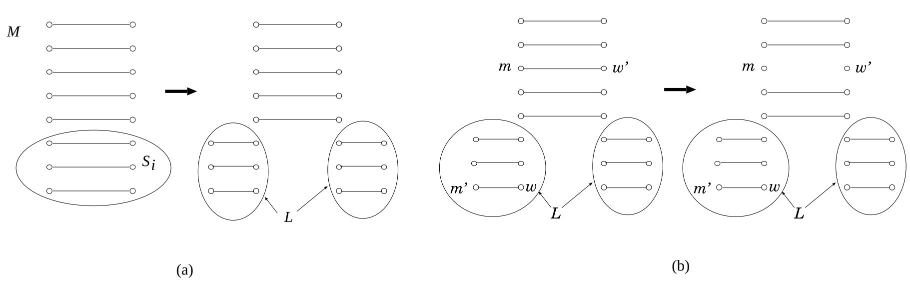

Fig. 2. Execution of Increase

4.1 Correctness of Increase

We give a sufficient condition for Increase to achieve a successful computation.

Lemma 4. *If S consists of only good edges, and if $|M| \leq \frac{|M_{opt}|}{2} + c \log N$, then there is at least one way of selecting i such that* Increase *succeeds.*

The proof of this lemma uses a series of lemmas. In the following lemmas, we assume assumptions in Lemma 4, namely, S consists of only good edges, and $|M| \leq \frac{|M_{opt}|}{2} + c \log N$, even if they are not explicitly stated in the statement of each of following lemmas.

Lemma 5. *There exists i^* such that, after executing the Gale-Shapley algorithm (at lines 5 and 6 of Fig. 3), every person in $S^m_{i^*} \cup S^w_{i^*}$ is matched with a partner who is at least as good as his/her partner in M_{opt}.*

Proof. Consider the following procedure. (Note that we consider this procedure only for the proof of this lemma. This procedure cannot be performed by algorithm Increase since it does not know M_{opt}.): Let S^m and S^w be sets of

Procedure INCREASE(M, S)

```
F^m := set of all single men in M;  F^w := set of all single women in M;
let S_1, S_2, ..., S_n be all subsets of S of size |S|/4;
for i := 1 to m
    {S_i^m := set of all men in S_i;  S_i^w := set of all women in S_i;
     Find a matching between S_i^m and F^w
              using the men-propose Gale-Shapley algorithm;
          (To do this, remove all persons not in S_i^m ∪ F^w from each person's list,
             and break all ties arbitrarily.)
     Find a matching between S_i^w and F^m
              using the women-propose Gale-Shapley algorithm;
          (To do this, remove all persons not in S_i^w ∪ F^m from each person's list,
             and break all ties arbitrarily.)
    if (∃p s.t. p ∈ S_i^m ∪ S_i^w and p remains single after the Gale-Shapley algorithm)
          exit for-loop;    /* the current i was not good choice */
     else
          {L := the set of all pairs obtained by the Gale-Shapley algorithm;
            M_i := M − S_i ∪ L;
            while (∃(m, w) ∈ BP(M_i) s.t. both m and w have a partner in M_i)
                      {if ( (m, M_i(m)) ∈ L and (M_i(w), w) ∈ L )
                           exit for-loop;    /* the current i was not good choice */
                        if ( (m, M_i(m)) ∈ M_i − L and (M_i(w), w) ∈ M_i − L )
                           exit for-loop;    /* the current i was not good choice */
                        if ( (m, M_i(m)) ∈ M_i − L and (M_i(w), w) ∈ L )
                           M_i := M_i − {(m, M_i(m))};
                        if ( (m, M_i(m)) ∈ L and (M_i(w), w) ∈ M_i − L )
                           M_i := M_i − {(M_i(w), w)};
                      }                   /* end while */
            if ( |M_i| > |M| )
                  output M_i and terminate;
            else exit for-loop;    /* the current i was not good choice */
            }                   /* end else */
     }                   /* end for */
output "error" and terminate;
```

Fig. 3. Procedure INCREASE

all men and women in S, respectively. Modify preference lists of all persons in $S^m \cup S^w \cup F^m \cup F^w$ in the same way as in the execution of INCREASE. Furthermore, in each man $m(\in S^m)$'s list, remove all women strictly below $M_{opt}(m)$. Similarly, in each woman $w(\in S^w)$'s list, remove all men strictly below $M_{opt}(w)$. It should be noted that for any person p in $S^m \cup S^w$, $M_{opt}(p)$ is in $F^m \cup F^w$ since any element of S is a good edge, and hence is not removed from p's list.

Apply the men-propose Gale-Shapley algorithm to the subinstance defined by S^m and F^w. It is not hard to see that at least half of S^m are matched at the termination of the Gale-Shapley. To see this, suppose the contrary, and let $A \subseteq S^m$ be the set of single men ($|A| > |S^m|/2$). Then, each man m in A is rejected by $M_{opt}(m)$. (Recall that $M_{opt}(m)$ is in m's list.) When $M_{opt}(m)$ rejected m, $M_{opt}(m)$ was matched with someone better than m, and during the execution of the Gale-Shapley algorithm, she never becomes single. So, at the termination, more than $|S^m|/2$ women are matched but this means that more than $|S^m|/2$ men are matched, a contradiction.

Now, if $m \in S^m$ has a partner after the execution of the Gale-Shapley algorithm, call m a *successful man*. Call a woman in S^w a *successful woman* if and only if her partner in M is a successful man (there are at least $|S|/2$ successful women). Now apply the women-propose Gale-Shapley algorithm to the subinstance defined by all successful women in S^w and F^m. If, in the resulting matching, a successful women gets a partner, call her a *super-successful woman*. For the same reason as above, at least half of all successful women are super-successful. Call a pair $(m, w) \in S$ a *super-successful pair* if and only if w is a super-successful woman. There are at least $|S|/4$ super-successful pairs.

Since $S_1, S_2, \cdots, S_n$ are all subsets of S with size exactly $|S|/4$, there exists at least one i such that S_i consists of only super-successful pairs. Let i^* be one of such i. It is not hard to see that after INCREASE completes the Gale-Shapley algorithm (of lines 6 and 7), each person in $S^m_{i^*}$ and $S^w_{i^*}$ is matched with at least as good a partner as one obtained by the above procedure. This completes the proof. □

In the following lemmas, i^* always denotes the one that satisfies the condition of Lemma 5.

Lemma 6. *M_{i^*} at line 11 of Fig. 3 satisfies following (1) and (2): (1) $|M_{i^*}| = |M| + \frac{k}{4}\log N$. (2) Consider an arbitrary blocking pair $(m, w) \in BP(M_{i^*})$ such that both m and w are matched in M_{i^*}. Then, exactly one of $(m, M_{i^*}(m))$ and $(M_{i^*}(w), w)$ is in $M_{i^*} - L$ and the other is in L.*

Proof. (1) Recall that $|S_{i^*}| = |S|/4 = \frac{k}{4}\log N$ and $|L| = 2|S_{i^*}|$. Then, $|M_{i^*}| = |M| - |S_{i^*}| + |L| = |M| + |S_{i^*}| = |M| + \frac{k}{4}\log N$.

(2) First, suppose that both $(m, M_{i^*}(m))$ and $(M_{i^*}(w), w)$ are in $M_{i^*} - L$. Observe that, by the construction of M_{i^*}, both of these two pairs are also in M. This means that $(m, w) \in BP(M)$, which contradicts the stability of M.

Next, suppose that both $(m, M_{i^*}(m))$ and $(M_{i^*}(w), w)$ are in L. We have four cases to consider: (i) $m \in F^m, w \in F^w$, (ii) $m \in S^m_{i^*}, w \in F^w$, (iii) $m \in F^m, w \in S^w_{i^*}$ and (iv) $m \in S^m_{i^*}, w \in S^w_{i^*}$.

Case (i): By the definition of F^m and F^w, both m and w are single in M. But since (m, w) forms a blocking pair for M_{i^*}, m and w write each other on their lists. This contradicts the stability of M.

Case (ii): By the assumption that (m, w) is a blocking pair for M_{i^*}, $w \succ_m M_{i^*}(m)$. Since $w \in F^w$, w stays in m's list when his list is modified to apply the Gale-Shapley algorithm. So during the execution of the Gale-Shapley algorithm at line 5, m proposed to w, but w rejected m, so $M_{i^*}(w) \succeq_w m$. Then (m, w) cannot block M_{i^*}, a contradiction.

Case (iii): Similar to Case (ii).

Case (iv): Since (m, w) is a blocking pair for M_{i^*}, $w \succ_m M_{i^*}(m)$ and $m \succ_w M_{i^*}(w)$. But by Lemma 5, $M_{i^*}(m) \succeq_m M_{opt}(m)$ and $M_{i^*}(w) \succeq_w M_{opt}(w)$. Then, $w \succ_m M_{opt}(m)$ and $m \succ_w M_{opt}(w)$, which means that (m, w) is a blocking pair for M_{opt}, a contradiction. □

The proof of Lemma 4 is completed by the following lemma, which guarantees the size of $|M_{i^*}|$ at line 22 of Fig. 3.

Lemma 7. *M_{i^*} at line 22 of* Increase *satisfies* $|M_{i^*}| > |M|$.

Proof. First of all, it should be noted that Increase never fails on i^* at lines 7 and 8 by Lemma 5. Also, during the execution of the while-loop on i^*, Increase never fails by Lemma 6 (2). By Lemma 6 (1), we know that $|M_{i^*}| = |M| + \frac{k}{4} \log N$. However, during the execution of the while-loop, some pairs may be removed from $M_{i^*} - L$, which may decrease the size of M_{i^*}. Note that all pairs in $M_{i^*} - L$ are pairs in M. In the following, we show that if a pair in $M_{i^*} - L$ is removed during the while-loop, then the pair must be a bad edge of M. If this is true, the number of removed pairs in the while-loop is at most $4c \log N$ by Lemma 2, and thus $|M_{i^*}| \geq |M| + \frac{k}{4} \log N - 4c \log N > |M|$. (Recall that $c < \frac{k}{16}$.)

Suppose that during the while-loop of Increase, some pair is removed from M_{i^*}. Then, there is a blocking pair (m, w) for M_{i^*} and both m and w are matched in M_{i^*}. We have two cases: (1) $(m, M_{i^*}(m)) \in L$ and $(M_{i^*}(w), w) \in M_{i^*} - L$ (and hence $(M_{i^*}(w), w)$ is removed). (2) $(m, M_{i^*}(m)) \in M_{i^*} - L$ and $(M_{i^*}(w), w) \in L$ (and hence $(m, M_{i^*}(m))$ is removed). We consider only Case (1). (Case (2) can be treated similarly.) Now, suppose that the removed pair $(M_{i^*}(w), w)$ is a good edge of M. We will show a contradiction.

For Case (1), we further consider two cases: (1-1) $m \in F^m$ and (1-2) $m \in S^m_{i^*}$.

Case (1-1): Note that m is single in M since $m \in F^m$. Now observe that, as $(M_{i^*}(w), w) \in M_{i^*} - L$, w and $M_{i^*}(w)$ are matched in M, namely, $M_{i^*}(w) = M(w)$. Since $(m, w) \in BP(M_{i^*})$, it results that $(m, w) \in BP(M)$, which contradicts the stability of M. (In this case, we can have a contradiction without assuming that $(M_{i^*}(w), w)$ is a good edge of M.)

Case (1-2): Since we assume that $(M_{i^*}(w), w)$ is a good edge of M, $M(w) \succeq_w M_{opt}(w)$ by Lemma 1. For the same reason as above, $M_{i^*}(w) = M(w)$. So, $M_{i^*}(w) \succeq_w M_{opt}(w)$. As (m, w) is a blocking pair for M_{i^*}, it results that $m \succ_w M_{i^*}(w) \succeq_w M_{opt}(w)$. Next, consider the man m which we assumed to be in $S^m_{i^*}$. By Lemma 5, $M_{i^*}(m) \succeq_m M_{opt}(m)$. Again, as (m, w) is a blocking pair for M_{i^*}, $w \succ_m M_{i^*}(m)$. So, $w \succ_m M_{i^*}(m) \succeq_m M_{opt}(m)$. Consequently, we have that (m, w) is in $BP(M_{opt})$, a contradiction. □

5 Procedure Stabilize(M_0)

Stabilize takes a matching M_0 and makes it stable without decreasing the size. Recall that for any blocking pair (m, w) for M_0, at least one of m and w is single in M_0. For a matching M, define $BP_{s,m}(M) \subseteq BP(M)$ to be the set of all blocking pairs (m, w) for M such that m is single in M and w is matched in M. Similarly, $BP_{m,s}(M)$ ($BP_{s,s}(M)$ and $BP_{m,m}(M)$, respectively) denotes the set of all blocking pairs (m, w) for M such that m is matched and w is single (both m and w are single, and both m and w are matched, respectively) in M. Define $BP_{-,s}(M) = BP_{m,s}(M) \cup BP_{s,s}(M)$. Fig. 4 shows the procedure STABILIZE.

Procedure STABILIZE(M_0)

1: while ($BP_{s,m}(M_0) \neq \emptyset$)
2: {select $(m, w) \in BP_{s,m}(M_0)$;
3: $w^* :=$ woman s.t. $(m, w^*) \in BP_{s,m}(M_0)$ and there is no $(m, w') \in BP_{s,m}(M_0)$ s.t. $w' \succ_m w^*$;
4: $M_0 := M_0 - \{(M_0(w^*), w^*)\} \cup \{(m, w^*)\}$;
5: }
6: while ($BP_{-,s}(M_0) \neq \emptyset$)
7: {select $(m, w) \in BP_{-,s}(M_0)$;
8: $m^* :=$ man s.t. $(m^*, w) \in BP_{-,s}(M_0)$ and there is no $(m', w) \in BP_{-,s}(M_0)$ s.t. $m' \succ_w m^*$;
9: **if** (m^* is matched in M_0)
10: $M_0 := M_0 - \{(m^*, M_0(m^*))\} \cup \{(m^*, w)\}$;
11: **else**
12: $M_0 := M_0 \cup \{(m^*, w)\}$;
13: }

Fig. 4. Procedure STABILIZE

5.1 Correctness of STABILIZE

Lemma 8. *Suppose that an application of line 4 of* STABILIZE *updates* M_0 *as follows.*

$$M_0' := M_0 - \{(M_0(w^*), w^*)\} \cup \{(m^*, w^*)\}.$$

Then, following (1) through (3) hold. (1) $M_0'(w^*) \succ_{w^*} M_0(w^*)$ *and for any* $w(\neq w^*)$, $M_0'(w) = M_0(w)$. *(2)* $|M_0'| = |M_0|$. *(3) If* $BP_{m,m}(M_0) = \emptyset$, *then* $BP_{m,m}(M_0') = \emptyset$.

Proof. (1) Since (m^*, w^*) is in $BP(M_0)$, $m^* \succ_{w^*} M_0(w^*)$. So, $M_0'(w^*) \succ_{w^*} M_0(w^*)$ because $M_0'(w^*) = m^*$. The latter part of (1) is trivial because, among all women, only w^* changed a partner.

(2) This is trivial.

(3) Observe that three persons changed the partner by updating from M_0 to M_0': w^* obtained a better partner, m^* became matched from single, and $M_0(w^*)$ became single from matched. So, any blocking pair arising by changing from M_0 to M_0' is associated with the man $M_0(w^*)$. Since $M_0(w^*)$ is single in M_0', any pair in $BP(M_0') - BP(M_0)$ is not in $BP_{m,m}(M_0')$.

Next, consider $(m, w) \in BP(M_0') \cap BP(M_0)$. Since $BP_{m,m}(M_0) = \emptyset$, at least one of m and w is single in M_0. Recall that only m^* changed the status from single to matched. So if $m \neq m^*$, $(m, w) \notin BP_{m,m}(M_0')$.

Now consider a blocking pair $(m^*, w) \in BP(M_0') \cap BP(M_0)$. If w was single in M_0, she is also single in M_0' and hence $(m^*, w) \notin BP_{m,m}(M_0')$. So assume that w was matched in M_0. In this case, both (m^*, w^*) and (m^*, w) were in $BP_{s,m}(M_0)$. So, both w^* and w were candidates for being matched with m^* in M_0'. But since w^* was selected, it must be the case that $w^* \succeq_{m^*} w$. Hence (m^*, w) cannot block M_0', leading to a contradiction.

We have shown that any element in $BP(M_0') - BP(M_0)$ and $BP(M_0') \cap BP(M_0)$ is not in $BP_{m,m}(M_0')$. This completes the proof. □

Lemma 9. *Suppose that an application of lines 10 and 12 of* Stabilize *updates* M_0 *as follows.*

(Line 10) $M_0' := M_0 - \{(m^*, M_0(m^*))\} \cup \{(m^*, w^*)\}$.
(Line 12) $M_0' := M_0 \cup \{(m^*, w^*)\}$.

Then, following (1) through (3) hold. (1) In case of executing line 10, $M_0'(m^*) \succ_{m^*} M_0(m^*)$ *(in case of executing line 12,* m^* *becomes matched in* M_0'*), and for any* $m (\neq m^*)$, $M_0'(m) = M_0(m)$. *(2)* $|M_0'| \geq |M_0|$. *(3) If* $BP_{s,m}(M_0) \cup BP_{m,m}(M_0) = \emptyset$, *then* $BP_{s,m}(M_0') \cup BP_{m,m}(M_0') = \emptyset$.

Proof. The proof is similar to that of Lemma 8 and will be omitted. □

Lemma 10. *Let* M' *be the output of* Stabilize. *Then* $|M'| \geq |M_0|$ *and* M' *is stable.*

Proof. Consider an application of line 4 of Stabilize. By Lemma 8 (1), at least one woman gets better off and all other women do not change the marital status. Since there are N women, each with a preference list of length at most N, the number of repetitions of the first while-loop is at most N^2. Let M'' be the matching just before Stabilize starts the second while-loop. Then $BP_{s,m}(M'')$ is empty. (This is the condition for Stabilize to exit from the first while-loop.) Since, $BP_{m,m}(M_0)$ is empty, we can show that $BP_{m,m}(M'')$ is empty by applying Lemma 8 (3) repeatedly. Combining these two facts, it results that $BP_{s,m}(M'') \cup BP_{m,m}(M'')$ is empty. Also, by Lemma 8 (2), $|M''| = |M_0|$.

Similarly as above, each application of line 10 or 12 would make men better off (Lemma 9 (1)), and hence the number of repetitions of the second while-loop is at most N^2. Since $BP_{s,m}(M'') \cup BP_{m,m}(M'') = \emptyset$, we can show that, $BP_{s,m}(M') \cup BP_{m,m}(M') = \emptyset$ using Lemma 9 (3) repeatedly. However, the termination condition of Stabilize says that $BP_{-,s}(M') = \emptyset$. Consequently, $BP(M')$ is empty and hence M' is stable. By Lemma 9 (2), $|M'| \geq |M''|$. So, $|M'| \geq |M_0|$. □

References

1. R. Bar-Yehuda and S. Even, "A local-ratio theorem for approximating the weighted vertex cover problem," In Analysis and Design of Algorithms for Combinatorial Problems, volume 25 of Annals of Disc. Math., Elsevier Science Publishing Company, Amsterdam, pp. 27–46, 1985.

2. P. Berman and T. Fujito, “On the approximation properties of independent set problem in degree 3 graphs,” *Proc. WADS 95*, LNCS 955, pp. 449–460, 1995.
3. T. Elde, “Assignment engine,” Senior Honours Project Report, Department of Computing Science, University of Glasgow, 2000.
4. D. Gale and L. S. Shapley, “College admissions and the stability of marriage,” *Amer. Math. Monthly*, Vol. 69, pp. 9–15, 1962.
5. D. Gale and M. Sotomayor, “Some remarks on the stable matching problem,” *Discrete Applied Mathematics*, Vol. 11, pp. 223–232, 1985.
6. D. Gusfield and R. Irving, “The Stable Marriage Problem: Structure and Algorithms,” MIT Press, Boston, MA, 1989.
7. M. Halldórsson, R. Irving, K. Iwama, D. Manlove, S. Miyazaki, Y. Morita and S. Scott, “Approximability results for stable marriage problems with ties,” *Theoretical Computer Science*, Vol. 306, pp. 431–447, 2003.
8. M. Halldórsson, K. Iwama, S. Miyazaki and H. Yanagisawa, “Randomized approximation of the stable marriage problem,” *Proc. COCOON 2003*, LNCS 2697, pp. 339–350, 2003.
9. M. Halldórsson, K. Iwama, S. Miyazaki and H. Yanagisawa, “Improved approximation of the stable marriage problem,” *Proc. ESA 2003*, LNCS 2832, pp. 266–277, 2003.
10. E. Halperin, “Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs,” *Proc. SODA 2000*, pp. 329–337, 2000.
11. R. Irving, “Stable marriage and indifference,” *Discrete Applied Mathematics*, Vol. 48, pp. 261–272, 1994.
12. R. Irving, “Matching medical students to pairs of hospitals: a new variation on an old theme,” *Proc. ESA 98*, LNCS 1461, pp. 381–392, 1998.
13. R. Irving, D. Manlove and S. Scott, “The hospital/residents problem with ties,” *Proc. SWAT 2000*, LNCS 1851, pp. 259–271, 2000.
14. K. Iwama, D. Manlove, S. Miyazaki, and Y. Morita, “Stable marriage with incomplete lists and ties,” *Proc. ICALP 99*, LNCS 1644, pp. 443–452, 1999.
15. M. Karpinski, “Polynomial time approximation schemes for some dense instances of NP-hard optimization problems,” *Proc. RANDOM 97*, LNCS 1269, pp. 1–14, 1997.
16. T. Le, V. Bhushan, and C. Amin, “First aid for the match, second edition,” McGraw-Hill, Medical Publishing Division, 2001.
17. D. Manlove, R. Irving, K. Iwama, S. Miyazaki, Y. Morita, “Hard variants of stable marriage,” *Theoretical Computer Science*, Vol. 276, Issue 1-2, pp. 261–279, 2002.
18. B. Monien and E. Speckenmeyer, “Ramsey numbers and an approximation algorithm for the vertex cover problem,” *Acta Inf.*, Vol. 22, pp. 115–123, 1985.
19. H. Nagamochi, Y. Nishida and T. Ibaraki, “Approximability of the minimum maximal matching problem in planar graphs,” Institute of Electronics, Information and Communication Engineering, Transactions on Fundamentals, vol.E86-A, pp. 3251-3258, 2003.
20. C.P. Teo, J.V. Sethuraman and W.P. Tan, “Gale-Shapley stable marriage problem revisited: Strategic issues and applications,” *Proc. IPCO 99*, pp. 429–438, 1999.
21. M. Zito,“Randomized techniques in combinatorial algorithmics,” PhD thesis, Dept. of Computer Science, University of Warwick, 1999.